

TD Java Avancé (INSA 2021-2022)

TD1 : Flux de données

Exercice1 :

Ecrire un programme en java qui permet de réaliser les taches suivantes :

1. Définir une classe Personne qui prend comme attributs : le nom, la date de naissance, le pays, le numéro de sécurité social, le domaine de la profession de la personne.
2. Définir un constructeur et des méthodes pour :
 - a. Lire et écrire les attributs à partir de la console
 - b. Demander à l'utilisateur de faire entrer les informations correspondantes à 5 personnes différentes.
 - c. Lire le fichier « Liste des metiers.txt »
 - d. Si les professions des 5 personnes n'est pas dans le fichier : Copier le fichier « Liste des metiers.txt » dans un nouveau fichier « maj - Liste des metiers.txt » (que vous devez créer en utilisant les classes et méthodes de java.io) et ajouter au fichier « maj Liste des metiers.txt » les nouvelles professions détectées sur de nouvelles lignes. Note : Utiliser un StringTokenizer pour lire les métiers qui sont sur la même ligne séparés par une virgule
 - e. Enregistrer les objets personnes créer dans un fichier « ObjetPersonne » en les sérialisant f. Ecrire une méthode qui déséréalise les objets enregistrés dans « ObjetPersonne » et les imprime
3. Ecrire votre main et instancier 3 nouvelles personnes et utiliser toutes les méthodes définies

Exercice 2 :

Programmez une classe qui attend 2 attributs un réel a et un entier b, et qui propose une méthode remplissant un fichier texte avec les multiples entiers de ce réel jusqu'à la valeur b selon le format suivant (dans cet exemple, a=0.5, et b =5) :

1 0.5

2 1

3 1.5

4 2

5 2.5

6 3

7 3.5

8 4

9 4.5

10 5

TD2 : Les événements et les écouteurs

Exercice :

Ecrivez 2 classes dans le même paquet :

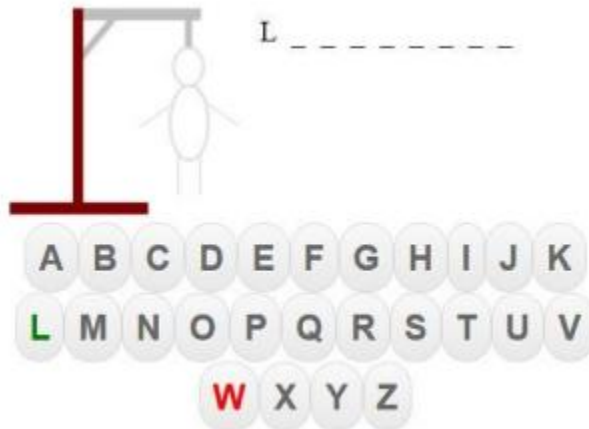
1. Classe Compétiteur (Nom, Prenom, Age, scoreTotal)
 - Dans le constructeur de cette classe initialiser la valeur du scoreTotal a 0
 - Définissez les setters et les getters
 - Définissez une méthode ajoutScore (int score) qui ajoute le score au scoreTotal
2. Classe Jeu (Compétiteur, DateDuJeu, score) et faites le nécessaire pour :
 - Score = 0 au début du jeu
 - Afficher 2 nombres aléatoires (choisis entre 1 et 10)
 - Une méthode qui attend pour 3 secondes la réponse de la multiplication du joueur : pour une réponse incorrecte (ou manque de réponse le score reste le même). Pour une réponse correcte score = score+1 et notifier le joueur de l'incréméntation de son score par 1 sans le rajouter au score Total
 - Un jeu se termine après 5 tours
 - Définissez une fonction qui change la valeur du score et le remet à zéro
 - Rajouter le score au score Total du joueur
 - Le joueur puisse s'opposer à une mise à jour du score supérieur à 5 ou inférieur à 0
 - Le joueur puisse notifier le jeu de s'arrêter en écrivant STOP dans la console au lieu de la réponse de multiplication et dans ce cas les
 - Le joueur doit être notifier à la fin du jeu un recevant un message de félicitation s'il a un score = 5

TD3 : les interfaces graphiques et les événements

Le but de cet exercice est de renforcer les idées et les concepts sur les événements dans les javas Beans et de les implémenter à travers une interface graphique.

Développez le jeu pendu en suivant les règles suivantes :

- 1) Créer une liste de mots dans un fichier
- 2) Choisissez un mot aléatoire du fichier (sans l'afficher)
- 3) Ouvrir une fenêtre graphique pour le jeu dans laquelle vous auriez à :
 - a. Définir 2 boutons
 - i. Bouton « Nouveau Mot » désactiver au début du jeu
 - ii. Bouton « Quitter » toujours activé et qui ferme la fenêtre quand on le clique
 - b. Indiquer le nombre de lettre du mot à deviner
 - c. Préparer un espace pour que le joueur fasse un choix lettre après lettre
 - i. Vous pourriez présenter un clavier de lettres comme une série de boutons et désactiver à chaque tour la lettre choisie
 - ii. Vous pourriez créer un espace texte où le joueur fait entrer exactement une lettre
 - d. A chaque fois une lettre est choisie deux cas apparaissent :
 - i. La lettre est présente dans le mot et donc sera afficher dans le mot caché (dans toutes les cases qui la représentent)
 - ii. La Lettre n'existe pas vous commencer/continuer à dessiner votre pendu dans cet ordre : Tête (Rond) -> Corps (Tiré vertical) -> Main droite -> main gauche -> pied droit -> pied gauche
 - e. Si le dessin du pendu est terminé afficher un message d'échec
 - f. Si le mot est deviné afficher un message de réussite
 - g. Activer le bouton « Nouveau Mot » à la fin de chaque tour de jeu qui va relancer la fenêtre avec un nouveau mot à deviner (étape 2-3)



TD4 : Programmation parallèle et concurrence

Exercice :

Dans un programme définissez :

- Deux threads : Le premier affiche les nombres entre 1 et 50 et Le deuxième affiche les nombres entre 1 et -50

- Une chaîne de caractère msg = « Début »

1. Qu'elle est votre observation ?
2. Changer le compteur dans le premier thread pour aller de 1 jusqu'à 1000. Que constatez-vous ? Exécuter le programme plusieurs fois et expliquer le comportement du programme
3. Afficher une phrase simple (insérer le code entre les appels à l'exécution des 2 threads) Que pourriez-vous dire ?
4. Trouver une solution pour que le deuxième thread se termine avant que le premier thread commence
5. Faites tous les changements nécessaires pour :
 - a. Changer la valeur de msg = « Deuxième thread s'est terminé en premier » si le deuxième thread est arrivé à la fin d'exécution avant le premier et interrompre l'exécution du premier thread. Si le deuxième thread s'est arrêté après le premier changer la valeur de msg = « Fin »
 - b. Afficher le message final.
6. Créer un troisième thread démon qui entre dans une boucle infinie. Expliquer l'effet du comportement de ce thread sur les 2 autres.
7. Créer un quatrième thread démon qui contenant une seule instruction comme l'affichage d'un message. Expliquer l'effet du comportement de ce thread sur les 2 premiers.
8. Changer les priorités des deux premiers threads (à vous de découvrir comment le faire et dans quel but vous voulez changer la priorité) analyser le résultat.
9. Pour le code en ligne faites le nécessaire pour avoir ces affichages :
 - a. **Affichage1 :**
5 10 15 20 25 100 200 300 400 500
 - b. **Affichage2 :**
5 10 15
100 200 300
20 25 30
400 500 600